

## CC3301 Programación de Software de Sistemas – Semestre Otoño 2019 – Tarea 1 – Prof.: Luis Mateu

**Parte a.-** Programe la función `concat_bits` declarada como:

```
typedef unsigned int uint; // enteros sin signo
uint concat_bits(uint x, int n, uint y, int m);
```

Si los bits de  $x$  son  $x_{31} x_{30} \dots x_1 x_0$  en donde  $x_0$  es el bit menos significativo y los de  $y$  son  $y_{31} y_{30} \dots y_1 y_0$ , la función `concat_bits` debe retornar  $0 \dots 0 x_{n-1} \dots x_0 y_{m-1} \dots y_0$ . Considere que  $n$  y  $m$  son menores que 32 y  $n+m \leq 32$ . Ejemplos de uso:

```
uint z1= concat_bits(0b101, 3, 0b11001, 5);
// z1=0b10111001
uint z2= concat_bits(0b11011, 3, 0x10101, 2);
// z2=0x01101
```

**Restricción:** Ud. no puede usar los operadores de multiplicación, división o módulo ( $*$  /  $\%$ ). Use los operadores de bits eficientemente.

**Parte b.-** Programe la función:

```
char *eliminados(char *str, char *pat);
```

Esta función entrega un nuevo string en donde se han eliminado del string `str` todas las apariciones del patrón `pat`. Ejemplos:

```
char *s1= eliminados("las palas van", "las");
// s1 es " pa van" (7 caracteres)
char *s2= eliminados("111011001", "10");
// s2 es "11101"
```

**Restricciones:** Ud. no puede usar el operador de subindicación `[ ]`, ni su equivalente `*(p+i)`. Para recorrer el string use aritmética de punteros. Use `malloc` para pedir memoria para el string resultante. Debe pedir exactamente la cantidad de bytes que necesita el resultado, no más. Para el primer ejemplo debe pedir 8 bytes, para el segundo 6 bytes.

**Parte c.-** Programe la función:

```
void eliminar(char *str, char *pat);
```

Esta función elimina del string `str` todas las apariciones del patrón `pat`. Ejemplos:

```
char s1[] = "las palas van"; // 13 caracteres
eliminar(s1, "las"); // s1 es " pa van" (7 caracteres)
char s2[] = "111011001";
eliminar(s2, "10"); // s2 es "11101"
```

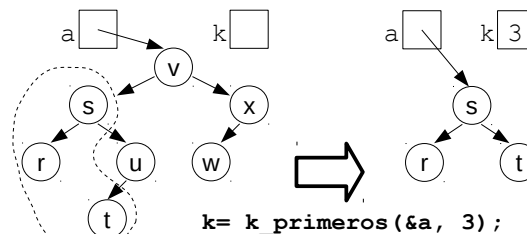
**Restricciones:** Ud. no puede usar el operador de subindicación `[ ]`, ni su equivalente `*(p+i)`. Para recorrer el string use aritmética de punteros. No puede

pedir memoria adicional ni declarar otros arreglos de caracteres. Use múltiples punteros para direccionar distintas partes del string. No puede usar la función de la parte b porque estaría pidiendo memoria adicional.

**Parte d.-** Programe la función:

```
int k_primeros(Nodo **pa, int k);
```

Esta función recibe un árbol de búsqueda binaria en `*pa` y lo poda dejando solo los primeros  $k$  nodos. El árbol resultante queda en `*pa` y debe ser un árbol de búsqueda binaria. La función retorna el número efectivo de nodos del árbol resultante. Si el árbol tiene menos de  $k$  nodos, entonces no se modifica y se retorna el número de nodos encontrados en el árbol. No cree nuevos nodos, reutilice los nodos existentes, por simplicidad no libere los nodos descartados. Ejemplo:



```
Hint: complete esta función
int k_primeros(Nodo **pa,
               int k) {
    Nodo *a= *pa;
    if (k==0) {
        *pa= ...
    }
    else if (a==NULL) {
        k= ...
    }
    else {
        int kizq=
            k_primeros(&a->izq, k);
        ...
    }
    ...
}
```

## Recursos

Baje `t1.zip` de U-cursos y descomprímalo. El directorio `T1` contiene los archivos `test-t1.c` que prueba si su tarea funciona, `t1.h` que incluye los encabezados de las funciones pedidas y `Makefile`. Ud. debe programar las 4 funciones pedidas en el archivo `t1.c`. El archivo `t1.c.plantilla` muestra los includes que Ud. debe agregar en `t1.c`. Lea las instrucciones que aparecen al comienzo de `Makefile`. Ud. debe usar el comando `make` para compilar su tarea. El programa de prueba lo felicitará si su tarea aprueba todos los tests o le indicará cuál test falla.

## Entrega

Ud. solo debe entregar el archivo `t1.c` por medio de U-cursos. Entregue su tarea solo si compila sin arrojar warnings en la máquina `anakena.dcc.uchile.cl` y aprueba todos los tests. Se crearán cuentas en `anakena.dcc.uchile.cl` para los alumnos que no son del DCC. Los alumnos del DCC ya tienen cuenta en ese computador. Se descontará medio punto por día de atraso. No se consideran los días sábado, domingo o festivos.